

NOTES FOR



NUMBER 23

December 1980

PUBLISHER

Ron Jeffries

EDITORIAL DIRECTOR

Glenn Fisher

EDITOR

Nancy Rhodes

ASSISTANT EDITORS

Diane Grove

Randy Lum

CIRCULATION

Jann Jeffries

Lori McFadden

Published by:

**The Code
Works™**

Box 550
Goleta, California 93116
805-683-1585

© 1980 The Code Works
All Rights Reserved

A CURSORY GLANCE

Yet another version of Basic is now available from Commodore. And that's not all: there is a new version of the disk operating system too! Since we've only had these new ROMs about two weeks, this will not be an in-depth review. But I will pass along some of our initial reactions.

As I understand things, there are now three flavors of Basic for the PET. (Yes, I know that some of these white boxes with electronics stuffed inside are called CBM's by Commodore. But they are all PETs to me. I'll never understand why they switched names. When is the last time your favorite brand of beer or whiskey or even - yeech - cigarette, changed it's widely recognized brand name?) Back to ROMs: in the beginning God and Chuck Peddle created the PET, with a small assist from Microsoft, who wrote the Basic. Those, my friends, were the so-called OLD ROMs. They had nasty surprises such as the fact that arrays couldn't be bigger than 256 bytes long. When you turned on your white box it smiled and politely announced "*** COMMODORE BASIC ***". (Just so you wouldn't get confused and think that you were using a Radio Shack or some other unspeakable horror.) Well, before long ye olde Commodore started selling what came to be known as NEW ROMs. In their infinite wisdom, they chose to give only a subtle hint as to which version you were using. This was done by having the machine print "### COMMODORE BASIC ###" when you turned it on. These NEW ROMs were a mixed blessing, especially for software vendors that had already published a bunch of programs for the OLD ROMs. The main problem is that Commodore reorganized what 6502 folks call "zero page", which gets used by most machine language programs.

The next thing you know, the 2040 floppy disk was released. Well, Basic and the Commodore operating system didn't know a darn thing about disks, and so interacting with the disk was rather clumsy. What to do? You guessed it: develop yet another set of ROMs with a Basic that knows that "DLOAD" means load a file from the disk, etc. etc.

But, there were some nasty bugs in the original DOS for the floppy, so it also had to be replaced! As you might guess, none of this is free. (Not that I'd expect it to be.) These goodies cost about \$125 for Basic 4.0 and another \$100 for DOS 2.1.

Well, the newest Basic is called "Version 4.0", and it now tells you the version when the machine is turned-on. (My congratulations to the Cheft!) In conjunction with DOS 2.1, you have a much nicer programming environment. (Especially if you have problems with "garbage collection". This happens when you are almost out of memory and are moving strings around. Before version 4.0, your machine would just go to sleep for a few minutes while it repacked memory. This has been fixed.) If you are a TOOLKIT fanatic like me, you'll need to exchange your old Toolkit ROM for the new TK-4.0 version. See your dealer or write the friendly folks at Palo Alto ICs, 430 Sherman Ave., Palo Alto CA 94306. They charge a reasonable fee for the swap. The reference manual for Basic 4.0 is the best yet from Commodore. (I secretly wonder if maybe they had Adam Osborne and crew write it...) But, the manual for the various flavors of floppy drives is wretched! For one thing, they made the mistake of trying to describe both the old DOS and the new DOS in the same book. In their defense I will say that they call it a "Preliminary Issue". Regardless of the poor documentation, the Commodore DOS is the most sophisticated DOS available for any of the mass market computers. I'm not sure how many people take advantage of the power of the DOS, but you have far more control of the system than is possible on - say - a Radio Shack, Apple or Atari.

CURSOR 23 HAS THESE PROGRAMS: (Program names ending with '!' use CB2 sound)

COVER23	Consider the spider... Graphics by Rob van Gelder
RECIPE	Convert your favorite recipe so you can feed an army. By Nancy Rhodes.
AMBUSH!	Isolate your enemy, and don't get trapped. By Phil Bayman.
ORRERY	The solar system in action on your PET. By Karl and Ernest Marhenke.
ENIGMA	Your very own encryption machine. By Alex Breed.
MWHIZ!	Quick! What's two plus two? By C.T. Nadovich.

MORE ABOUT THE PROGRAMS

RECIPE... This month, some new things are cooking at CURSOR. One of them is called Recipe, and it's for anyone who cooks, whether your specialty is chateau briande or toasted cheese and tomatoes. You'll especially appreciate Recipe if you're cooking for an army, and the recipe serves two, or you're cooking for two and the recipe feeds an army! Recipe will take any recipe and adjust it to make any amount, or to serve any number of people that you specify. All that you need to do is type in your recipe, how much it makes, and how much you'd like it to make. Let Recipe do the rest.

Each ingredient should be entered with the quantity first, expressed as a whole number and a fraction (no decimals), then the unit (ie. cups, tsps, oz.), followed by the ingredient. Each ingredient should be typed on a new line, and when you are through typing in the recipe, hit the RETURN key again.

When the PET asks you how much your recipe makes, type in the amount. For example, you might answer: 5 cups, 3 dozen, 2 (serves two people) etc. When the PET asks you how much you would like the recipe to make, answer in the same way (15 quarts, 1 1/2 dozen, 50, etc.).

If the Pet can't understand a line, it will show you the line and tell you what it thinks is wrong. It will then have you type in the line again. Keep in mind the fact that the Pet makes no attempt to interpret what you type until it tries to do the conversion. Thus, it won't report errors while you're typing. Sigh! Those of you with printers may have the Pet print the converted recipe on the printer, for permanent reference. Those of you without printers will have to lug the Pet to the kitchen. That's all there is to it. Now go try out your favorite recipe (and send us a batch!).

AMBUSH! A game of strategy in which you, as general, try to outmaneuver your opponent, the enemy general. The object of Ambush is to isolate the enemy by destroying the forest around their troops. The two generals take alternate turns, and each turn consists of a move and an attack. The move is made to any of the eight adjacent forests, (provided it has not been destroyed yet), by typing the number of the forest. Bombs are also sent in any of the eight directions, and may travel any distance. Please note that when you move or when you attack that the move or attack must be in a direct straight line from your current position. This simply means that legal moves are: up, down, left, right, and any of the diagonals. You specify your target by typing in the number of the forest you wish to bomb. You may not bomb the opponent directly, but you may destroy an adjacent forest.

Play continues on in this manner until one general is surrounded by forest that has been destroyed and thus can not move. (Unlike the real thing, you can't jump in a helicopter and escape to the pleasures of Saigon at the last minute.) The trick is to not let yourself be cornered (not as easy as it looks!). Good Luck.

ORRERY... (Note for 8K PET users: Orrery can be loaded into an 8K PET, but you'll have to remove the directions (lines 62100 to 62210) before you can run the program!) While the name Orrery might sound a bit cryptic, it actually describes our program precisely. An "orrery" is an animated model of the Sun and planets. Traditionally, an orrery is a mechanical marvel with intricate systems of gears to perfectly simulate the revolutions of the planets. Now, with the advance of technology, your PET becomes an orrery.

Orrery will ask for a year, which must be between 1680 and 2280, and a month, which may be entered as a number or by name. It will then present a view of the Sun and the planets from Mercury through Jupiter as they appear on the first day of the month chosen. Every few seconds, the date is changed and the view is updated.

The letters V, S, A, and W around the edge of the diagram indicate the directions from the Sun to the Vernal Equinox, the Summer Solstice, the Autumnal Equinox, and the Winter Solstice. Once Orrery starts running, you can give it commands to control it. Not all nine planets can be seen at once, since the range of planetary distances is 120:1. Pressing O (Oh, not zero) will switch to the "outer" planets (Jupiter on out to Pluto). To switch back again, press I for "inner".

The time interval between displays can be changed to 30 days by pressing M (for Medium), and to 180 days (better for the Outer planets) by pressing F (for Fast). Pressing S (Slow) returns you to a 7 day interval. Pressing R (Retrograde) will make the Solar System run backwards (such awesome power at your fingertips!). Pressing D (Direct) will make it go forward again. To freeze the planets in their current view, type H (for Halt). Pressing another key will start them going again. N (for New) allows a new date to be entered, and Q will end Orrery.

ENIGMA... Enigma allows you to encipher and decipher messages using a system based on the German World War II code machine named (oddly enough) Enigma. It may be a help to anyone who has a 'top-secret' message to send to a friend with a PET, or anyone with some confidential information to save.

At the heart of the machine are five interchangeable 'code wheels', of which three are installed in the machine at any one time. Each wheel can be set at one of twenty-six positions. The chosen wheels and their settings determine the code. You may set the code wheels yourself, or let the PET do it randomly. If you choose to do it yourself, you must tell the PET which three of the five wheels go into the slots, and indicate the settings for each of these three wheels (by the letters A-Z).

Encoding and decoding are done in exactly the same way. Type in the message (be it encoded or not), followed by a blank line. The blank line tells the Pet that it can run the message through the code machine, which it does. As it does this, it prints out the result one letter at a time. If the typed message was encoded, the original message (or 'clear text' as it's called in the code biz) will appear. If you typed in clear text, the encoding will be printed. (Speaking technically, the cipher that the code machine uses is its own inverse.) One limitation on the encoding process: only letters are encoded; numbers, punctuation, and spaces are thrown away. Since the resulting message must consist of blocks of five letters, there may be meaningless letters attached to the end of your message. For example, if you type in HI and the coded version is SKFIM, and you then type in SKFIM, you may get back something like HIDSQ. The extra letters are just to pad the message out to an even group of five letters. (The message is made exactly a multiple of five characters to fool the enemy.)

MWHIZ! If you enjoy solving arithmetic problem quickly, you'll find this two-person program to be a lot of fun. A problem is presented on the screen, along with a proposed answer. The gimmick is that the answer may or may not be correct for that problem. The two players each have a button to indicate that the answer is correct, as well as one to say that the answer is wrong. Your score depends on two things: did you correctly indicate whether the answer was correct or not, and how fast did you do so. If you are right, your total score goes up. But if you are wrong - well, your score goes down. If you are right and you answer very quickly, you might get as much as 13 points. The longer it takes you to answer, the less you get. For example, if you answer within 2.4 seconds you get 5 points. (But you'll never receive less than one point!). You are allowed about ten seconds to answer, at which point the machine tells you what the answer was, and neither players score is affected.

In line 190 there is a DATA statement that can be modified so that the relative frequency of addition, subtraction, multiplication and division can be specified. On your Cursor tape, the four numbers are all "1". But to change the program so that it only presents addition problems, you would change line 190 to: 190 DATA 1,0,0,0. Other modifications you may want to make are to change the highest possible score for an answer, which is controlled by the equation in line 110. Change the number "12" in that line to the maximum score minus one that you want to get for a very fast answer. If you'd like the game to end at, say, 100 points (instead of 50), change line 100 so that MP=100.